

CLAIMS

1. A method comprising:
identifying that a storage array is close to permanently losing data; and
giving, in response to identifying that the storage array is close to permanently losing data, input/output (I/O) requests for rebuilding at least a portion of the storage array priority over host I/O requests.
2. A method as recited in claim 1, wherein the identifying comprises identifying that the storage array is close to permanently losing data when failure of one additional storage device of a plurality of storage devices in the storage array would result in permanent data loss in the storage array.
3. A method as recited in claim 1, wherein the storage array comprises a redundant array of independent disks (RAID) system.
4. A method as recited in claim 3, wherein:
the RAID system includes a plurality of RAID levels;
the identifying comprises identifying when at least one of the plurality of RAID levels is close to permanently losing data; and
giving, rebuild I/O requests priority over host I/O requests only for the at least one RAID level that is close to permanently losing data.
5. A method as recited in claim 4, wherein one of the plurality of RAID levels includes RAID level 6.

6. A method as recited in claim 1, further comprising giving host I/O requests priority over rebuild I/O requests if the storage array is not close to permanently losing data.

7. A method as recited in claim 1, wherein giving I/O requests for rebuilding at least a portion of the storage array priority over host I/O requests comprises:

placing both I/O requests for rebuilding at least the portion of the array and host I/O requests into a queue in the order they are received; and

processing the I/O requests for rebuilding and the host I/O requests from the queue in a first-in-first-out (FIFO) manner.

8. A method as recited in claim 1, wherein giving I/O requests for rebuilding at least a portion of the storage array priority over host I/O requests comprises:

allocating, among a plurality of resources in the storage array and a corresponding controller, more resource usage to the I/O requests for rebuilding than to the host I/O requests.

9. A method as recited in claim 1, wherein giving I/O requests for rebuilding at least a portion of the storage array priority over host I/O requests comprises preempting a host I/O request in favor of a rebuild I/O request.

10. A method as recited in claim 1, wherein how many failed disks in the storage array can be endured without permanently losing data varies based at least in part on a particular redundant array of independent disks (RAID) architecture level of the storage array.

11. One or more computer-readable media having stored thereon a computer program that, when executed by one or more processors of a computer, causes the one or more processors to perform acts including:

identifying that a storage array is close to permanently losing data; and

giving, in response to identifying that the storage array is close to permanently losing data, input/output (I/O) requests for rebuilding at least a portion of the storage array priority over host I/O requests.

12. One or more computer-readable media as recited in claim 11, wherein the identifying comprises identifying that the storage array is close to permanently losing data when failure of one additional storage device of a plurality of storage devices in the storage array would result in permanent data loss in the storage array.

13. One or more computer-readable media as recited in claim 11, wherein giving I/O requests for rebuilding at least a portion of the storage array priority over host I/O requests comprises:

allocating, among a plurality of resources in the storage array and a corresponding controller, more resource usage to the I/O requests for rebuilding than to the host I/O requests.

14. One or more computer-readable media as recited in claim 11, wherein how many failed disks in the storage array can be endured without permanently losing data varies based at least in part on a particular redundant array of independent disks (RAID) architecture level of the storage array.
15. An apparatus comprising:
a priority identifier to determine whether host input/output (I/O) requests or rebuild I/O requests for a storage array are to have priority; and
a request dispatcher, communicatively coupled to the priority identifier, to select host I/O requests and rebuild I/O requests for execution based at least in part on whether host I/O requests or rebuild I/O requests are to have priority.
16. An apparatus as recited in claim 15, wherein the storage array comprises a redundant array of independent disks (RAID) system.
17. An apparatus as recited in claim 15, further comprising a request queue structure into which the rebuild I/O requests and the host I/O requests are placed to await selection for execution by the request dispatcher.
18. An apparatus as recited in claim 17, wherein the request dispatcher is configured to select requests from the top of the queue structure regardless of whether the requests are host I/O requests or rebuild I/O requests.

19. An apparatus as recited in claim 17, further comprising a queue controller, communicatively coupled to the request queue structure, configured to order requests in the queue structure so that host I/O requests are higher than rebuild requests only if host I/O requests are to have priority.
20. An apparatus as recited in claim 17, wherein the request queue structure includes a plurality of queues.
21. An apparatus as recited in claim 15, further comprising:
a plurality of resources; and
wherein the request dispatcher is to limit the host I/O request usage of at least one of the plurality of resources if rebuild I/O requests are to have priority.
22. An apparatus as recited in claim 15, wherein the priority identifier is to determine that rebuild I/O requests are to have priority if failure of one additional storage device of a plurality of storage devices in the storage array would result in data loss in the storage array.
23. An apparatus as recited in claim 15, further comprising a request processor, communicatively coupled to the request dispatcher, to process I/O requests and preempt a host I/O request in favor of a rebuild I/O request.